A General Framework for Adaptive and Online Detection of Web attacks

Wei Wang , Florent Masseglia
Project AxIS, INRIA Sophia Antipolis
06902 Sophia Antipolis, FRANCE
wwangemail@gmail.com

Thomas Guyet, René Quiniou,
Marie-Odile Cordier
Projet DREAM, INRIA Rennes/ IRISA
35042 Rennes, FRANCE

ABSTRACT

Detection of web attacks is an important issue in current defense-in-depth security framework. Many existing anomaly detection methods require a large amount of precisely labeled data to build a static model that is then used for attack detection. In practical environments, however, labeled data is very difficult to obtain. Moreover, the audit data for attack detection is typically streaming and the behavioral model is always evolving. Static detection models thus lead to considerable false positives. In this paper, we propose a novel general framework for adaptive and online detection of web attacks. The general framework can be based on any online clustering methods. A detection model based on the framework is able to learn online and deal with concept drift in web audit data streams. Str-DBSCAN that we extended DBSCAN [1] to streaming data as well as StrAP [3] are both used to validate the framework. The detection model based on the framework automatically labels the web audit data and adapts to normal behavior changes while identifies attacks through dynamical clustering of the streaming data. A very large size of real HTTP Log data collected in our institute is used to validate the framework and the model. The preliminary testing results demonstrated its effectiveness and efficiency.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*

General Terms

Design, Experimentation, Measurement, Security

Keywords

Anomaly Detection, Intrusion Detection, Clustering

1. INTRODUCTION

Anomaly intrusion detection is a widely studied topic in computer networks because of its capability of detecting novel attacks. Anomaly detection normally defines a profile of a subject's normal activities and attempts to identify any unacceptable deviation as possibly the result of an attack. Many existing anomaly IDSs (Intrusion Detection System) have some difficulties for practical use.

First, a large amount of precisely labeled data is very difficult to obtain in practice. Precise labeling is a very hard task and sometimes the data is even forbidden to be provided to security experts for labeling due to some privacy policies. In contrast, many existing anomaly detection approaches need precisely labeled data to train the detection model. Second, data for intrusion detection is typically steaming and the detection models should be frequently updated with new incoming labeled data. However, many existing anomaly detection methods involve off-line learning, where data is collected, manually labeled and then fed to a learning method to construct normal or attack models. Quickly and manually labeling the data is difficult and thus it is quite expensive to frequently re-train the IDS with new clean labeled data. Third, many current anomaly detection approaches assume that the data distribution is stationary and the model is static accordingly. In practice, however, data involved in current network environments always evolves. An effective anomaly detection method, therefore, should have adaptive capability to deal with the "concept drift" problem. That is, the model should be automatically updated to adapt to normal behaviors when there is a change detected.

2. THE FRAMEWORK

Adaptive and online anomaly attack detection is confronted with three important issues: (1) how to automatically and precisely identify the anomalies in massive audit data streams on the condition that no labeled data is available to learn or to train; (2) how to automatically select as much as new incoming clean normal data to update the model without incorporating the abnormal data; (3) how to update the detection model to adapt to behavioral changes in data streams. Our framework addresses these issues through online and unsupervised clustering algorithms in data streams, under the assumption that normal data is very large while abnormal data is rare in practical detection environments. With this assumption, cluster sizes of normal data are large, whereas abnormal data lies in a sparse region of the input space and their corresponding cluster sizes are relatively small. We then use the size as well as looseness of each cluster to identify the anomalies. Our method adaptively detects attacks with following three steps (the pseudo code is described in Fig. 1).

- **Step 1.** Building the initial model with some online clustering algorithms. The first bunch of data is clustered and the exemplars (or cluster centers) as well as their associated items are thus obtained. Some outliers are identified, marked as *suspicious* and then put into a reservoir.
- **Step 2.** Identifying outliers and updating the model in the data streaming environment. As the audit data

stream flows in, each incoming data item is compared to the exemplars. If too far from the nearest exemplar, the item is identified as an outlier, marked as *suspicious* and then put into the reservoir. Otherwise the item is regarded as *normal* and the model is updated accordingly with the normal item.

- **Step 3.** Rebuilding the model and identifying attacks. The model rebuilding criterion is triggered if the number of incoming outliers exceeds a threshold or if a time period is up to another threshold. The detection model is rebuilt with the current exemplars and the outliers in the reservoir, using the clustering algorithm again. An *attack* is identified if an outlier in the reservoir is marked as suspicious once again after the model rebuilding.

```
Audit data stream $x_1, \dots x_t, \dots$; fit threshold $N, \epsilon$
Clustering $(x_1, \dots, x_T)$ with some clustering algorithms
  $e_i$ is the exemplar (clustering center) of one cluster
  $n_i$ is the number of items in exemplar $e_i$
  $\mu_i$ is the mean sum of the distances between each exemplar $e_i$ and its corresponding items
Reservoir = {}
if $n_i \le N$ or $\mu_i \ge \epsilon$ then
    Reservoir $\leftarrow$ all items $x_j$ in $e_i$
end if
for $t > T$ do
  find $e_i$ which is the nearest exemplar to item $x_t$
  if $d(e_i, x_t) < \epsilon$ then
    Update model
  else
    Reservoir $\leftarrow x_t$
  end if
  if change detected then
    Rebuild the model (Re-clustering)
    Consider all the exemplars $e_j$ in Reservoir
    if $e_j$ appears at least twice in Reservoir and ($n_j \le N$ or $\mu_j \ge \epsilon$) then
        all the items in $e_j$ are attacks
    else
        Update the model
    end if
  end if
end for
```

Fig.1. Pseudo code of the framework

3. DETECTION MODELS BASED ON THE FRAMEWORK

The detection models can be based on any online clustering algorithms. In this paper, we extend DBSCAN [1] to Str-DBSCAN that is suitable for clustering streaming data. The Str-DBSCAN as well as a newly invented StrAP [3] are both used to build the detection models based on the framework, because these two clustering algorithms have no need to define the number of clusters beforehand.

DBSCAN is a density based clustering algorithm. After the initial clustering, each cluster is represented by an exemplar that is closest to its center. In data streaming environments, upon a "concept change" has been detected, Str-DBSCAN clusters all the current exemplars as well as the outliers that are the points far from the exemplars. During the clustering, we continually update the exemplars with some weights, so that some exemplars will be forgotten if they seldom appear in a period while some exemplars will be strengthened if they appear very frequently.

Affinity Propagation (AP) is a recently developed clustering algorithm and Zhang et al. extended it to StrAP in data steaming environments. AP clusters an initial data set and finds some exemplars to represent each cluster. In streaming environments, similarily StrAP continually updates the clusters and deal with "concept drift" in the data streams.

4. EXPERIMENTS AND CONCLUSION

In the experiments, we collected a very large data set of HTTP logs on the main Apache server of our institute for web attack detection. In order to increase the attacks, we downloaded 35 different types of attack from http://www.i-pi.com/HTTP-attacks-JoCN-2006 [2] and randomly inserted them into our data set.

To facilitate comparison, we also used k-NN to build a static model for intrusion detection. In the experiments, we set k=1 and compute the closest Euclidean distance between an incoming test vector X and each vector in the training data set. X is classified as anomalous if its closest distance is above a pre-defined threshold.

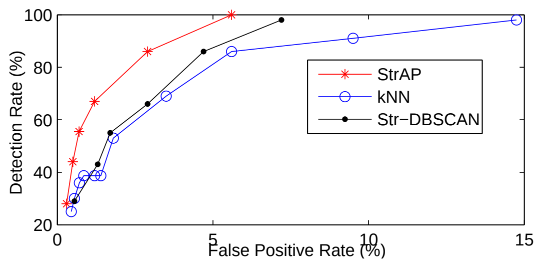

Fig.2. ROC curves with Str-DBSCAN, StrAP and k-NN

We use ROC curves (Detection Rates against False Positive Rates) to show the testing results shown in Fig. 2. The time cost is shown in Tab.1.

Table 1. CPU time used

StrAP	Str-DBSCAN	k-NN
633.8 s	2110.2s	6052.9s

It is seen from Fig.2 and Tab.1 that adaptive anomaly detection methods, Str-DBSCAN as well as StrAP, are more effective than static detection method, k-NN, because adaptive methods adopt to the behavioral changes while the static method does not. The adaptive methods are also effecient than static method because adaptive methods summarize the historical data into some simple concepts (e.g., exemplars) while the static method does not.

Web attack detection is becoming important as Web-based vulnerabilities represent a substantial portion of the security exposures of computer networks. Our framework is effective to detect attacks in an online and adaptive fashion without a priori knowledge (e.g., data distribution as well as labeled information).

5. REFERENCES

[1] M. Ester. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996.

[2] K. Ingham and H. Inoue. Comparing anomaly detection techniques for http. In *RAID*, 2007.

[3] X. Zhang, C. Furtlehner, and M. Sebag. Data streaming with affinity propagation. In *ECML*, 2008.